Jet Propulsion Laboratory
California Institute of Technology

Cockatoo

Chatbot for Cybersecurity

Anthony Barrett, Shahrouz R Alimo, Brian Kahovec, Edward Chow

JPL POC: Anthony Barrett, Ph.D.
NASA / Jet Propulsion Laboratory
California Institute of Technology
818-393-5372
anthony.barrett@jpl.nasa.gov

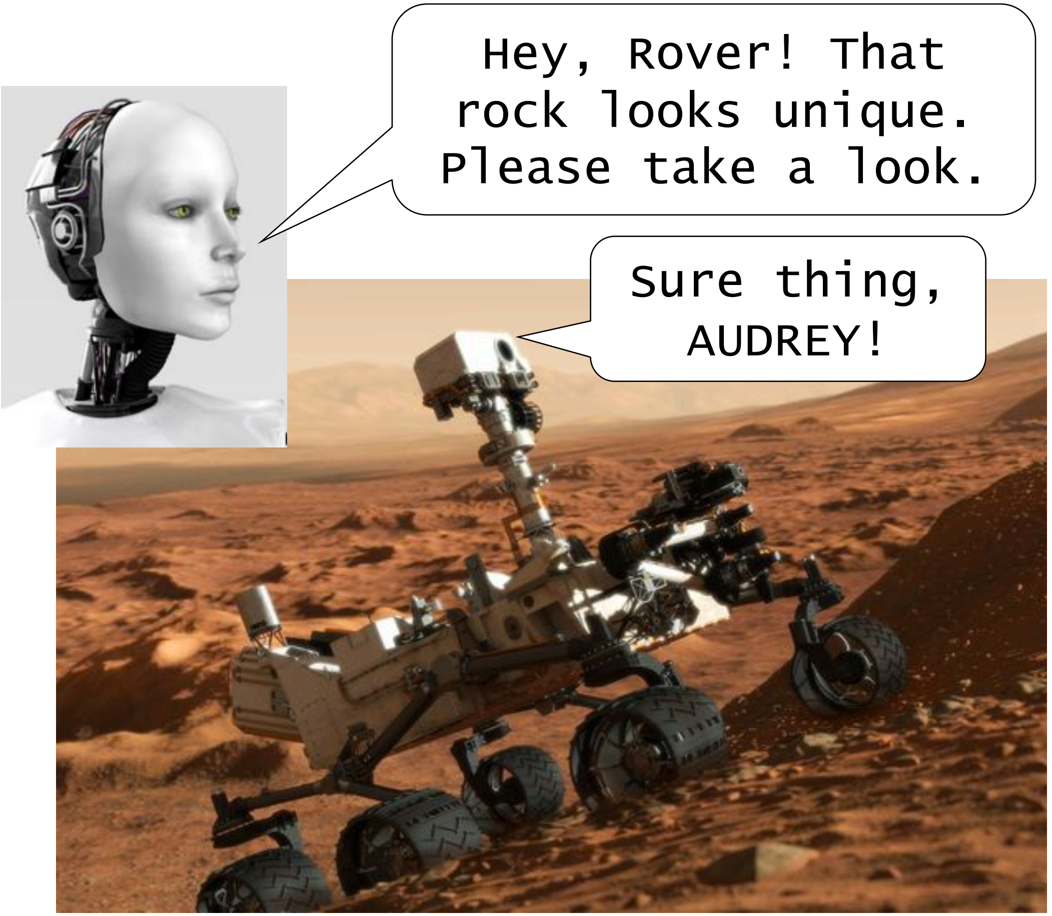
November 6, 2018

Outline

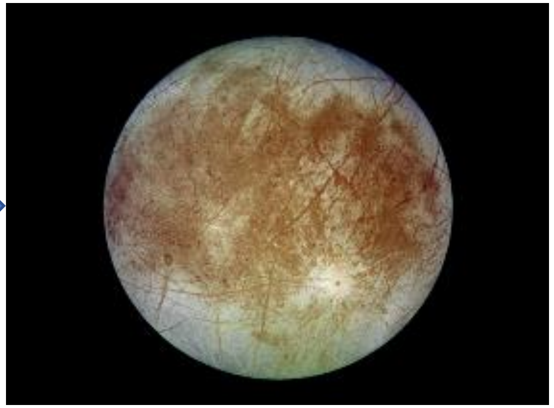
- AUDREY overview
- NLP enhancements for Chatbot UI
- Autoencoder enhancements to analyze logs
- K-means enhancements to analyze logs
- Future work

NASA Need for Next Generation Artificial Intelligence System

- ***AUDREY: Human-Like Autonomous System***

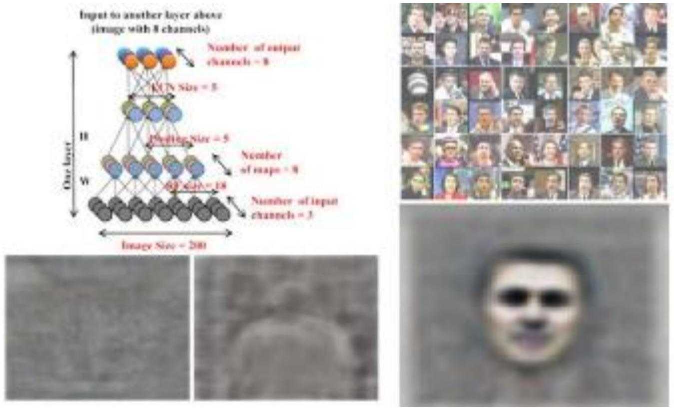


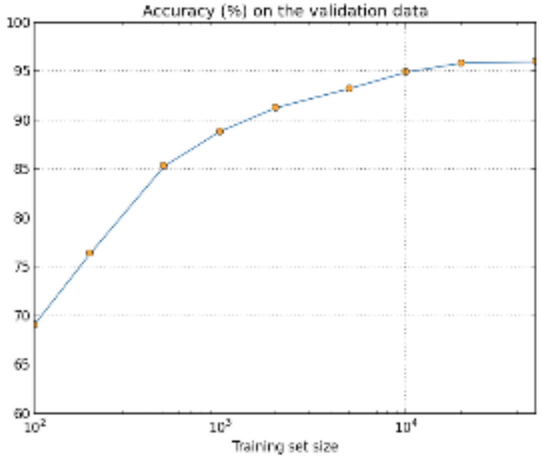
- × **Real-time Learning**
- × **Insight**
- × **Ingenuity**
- × **Work with Uncertainty**
- × **Train with Limited "Good" Data**



State-of-the-Art AI Technologies and Challenges Model Based, Deep Learning, or Human Expertise

- Traditional Rule-based AI difficulty with uncertainty
- Machine Learning techniques require experts to do feature engineering
- Deep Neural Network (DNN) needs a lot of training data



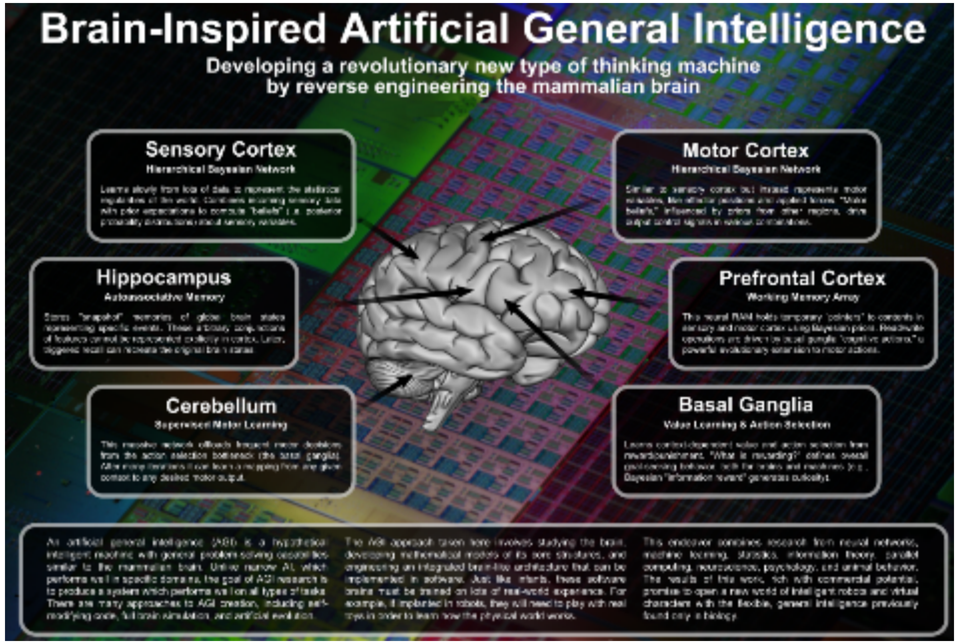


NASA needs a different approach

* Image source: google.com

AUDREY (Assistant for Understanding Data through Reasoning, Extraction, & sYnthesis)

- AUDREY uses bio-inspired Neural Symbolic Processing
  - Mixed neural and symbolic processing by achieving neural processing at symbolic level for higher level cognitive reasoning
- AUDREY leverages human intelligence to achieve better machine intelligence
- AUDREY capabilities:
  - Reasoning and learning new knowledge at the same time
  - Deal with missing or contradictory data
  - Automatically synthesize workflows to answer questions
  - Learn from human and a community of Audrey agents

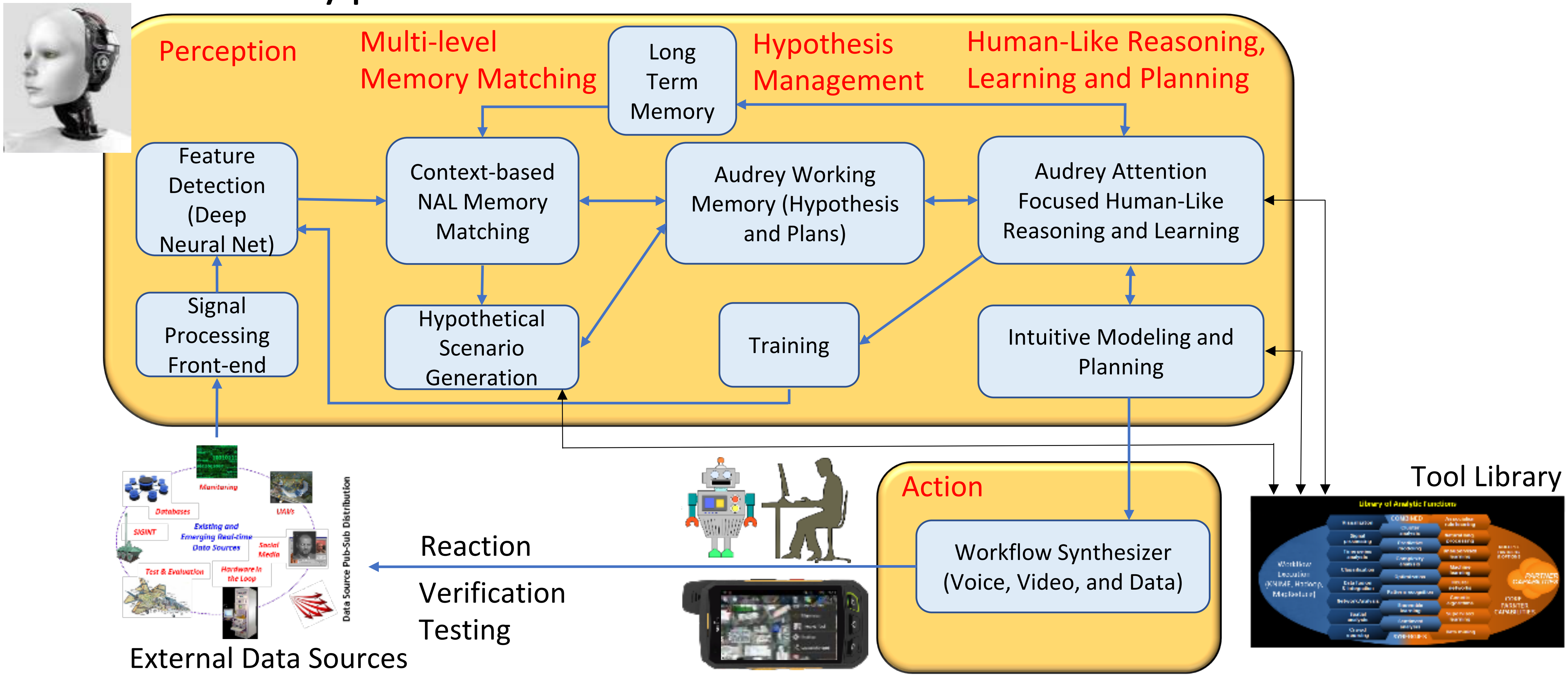


The Evolution of AI

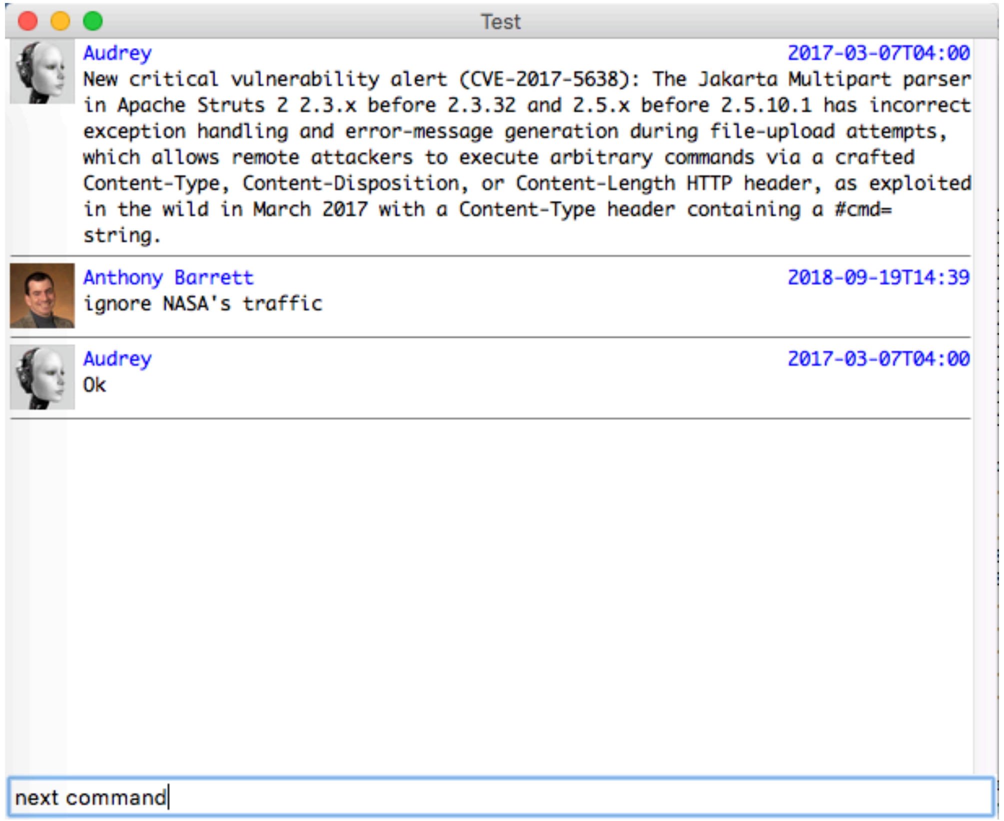
Achieves unprecedented levels of reasoning for previously unsolvable problems

* Image source: google.com

AUDREY: End-to-End Human-Like AI
DNN + Hypothesis + Generalization + Automation



NLP enhancements for Chatbot UI



- The current model is to interact textually with a user.
- Each analysis takes the form of a dialog, which can last for weeks.
- Each dialog results in a sequence of API calls.
- Workflows will be discovered from API call sequences, letting AUDREY anticipate requests.

Underlying NLP research

- Based on Combinatory Categorial Grammar (CCG)

Words	Category
netflows	N : $\lambda x.netflow(x)$
to	(N\N)/NP : $\lambda x.\lambda f.\lambda y.f(x) \wedge to(y,x)$
alpha7.jpl.nasa.gov	NP : 23.62.109.216
beta2.jpl.nasa.gov	NP : 23.62.106.239
…	…

- Forward and Backward Application
  - $X/Y : f \quad Y : a \quad \Rightarrow_{>} \quad X : f(a)$
  - $Y : a \quad X\backslash Y : f \quad \Rightarrow_{<} \quad X : f(a)$

Fast semantic parser in under 300 lines

List	traffic	to	alpha7.jpl.nasa.gov
S/N	**N**	**(N\N)/NP**	**NP**
$\lambda f.f$	$\lambda x.netflow(x)$	$\lambda y.\lambda f.\lambda x.f(y) \wedge to(x,y)$	23.62.109.216

N\N

$\lambda f.\lambda x.f(x) \wedge to(x, \text{23.62.109.216})$

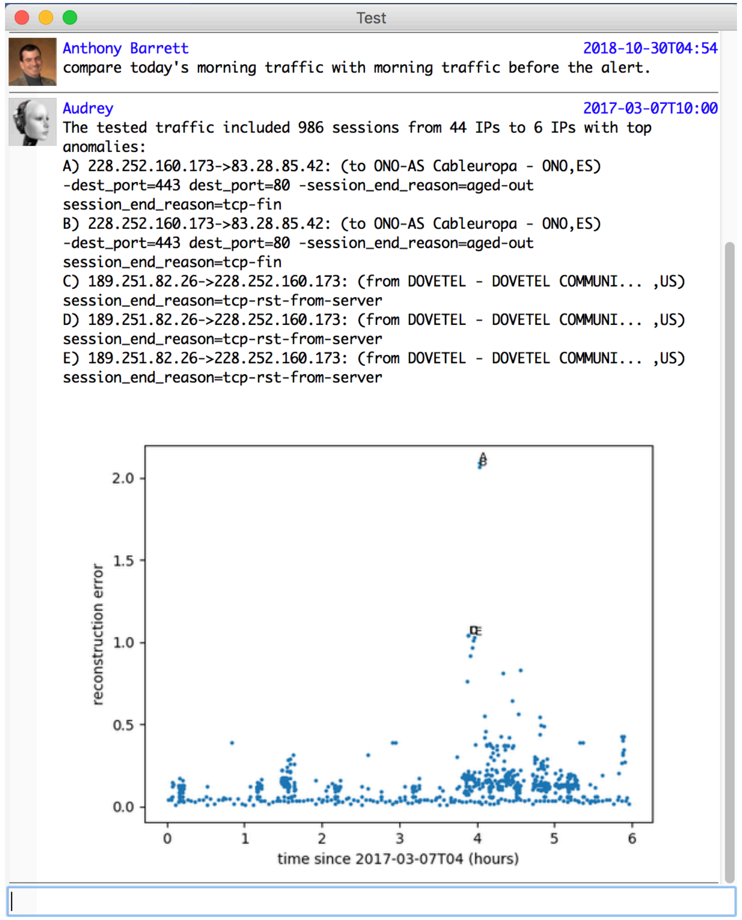
N

$\lambda x.netflow(x) \wedge to(x, \text{23.62.109.216})$

S

$\lambda x.netflow(x) \wedge to(x, \text{23.62.109.216})$

Autoencoder enhancements to analyze logs



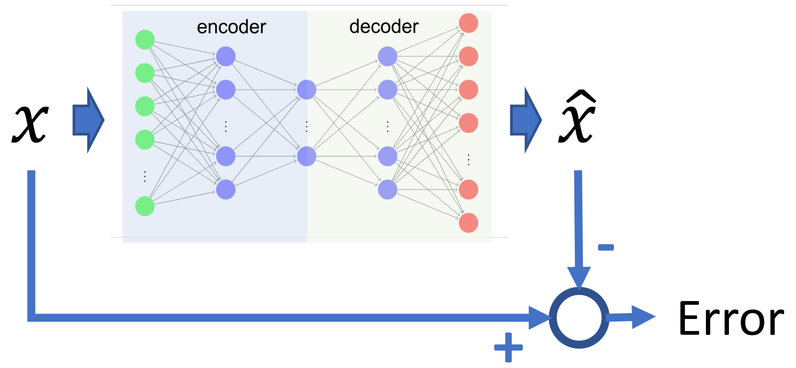
- A deep autoencoder (AE) is a special class of neural network that does not require labeled training data
  - Train AE on past traffic
  - Test new traffic against AE
- Unlike other NN solutions, autoencoders allow shallow inspection of "why"
  - Anomalous traffic exhibits high reconstruction error, which can be traced to specific symptoms.

Monitoring Traffic with an Autoencoder

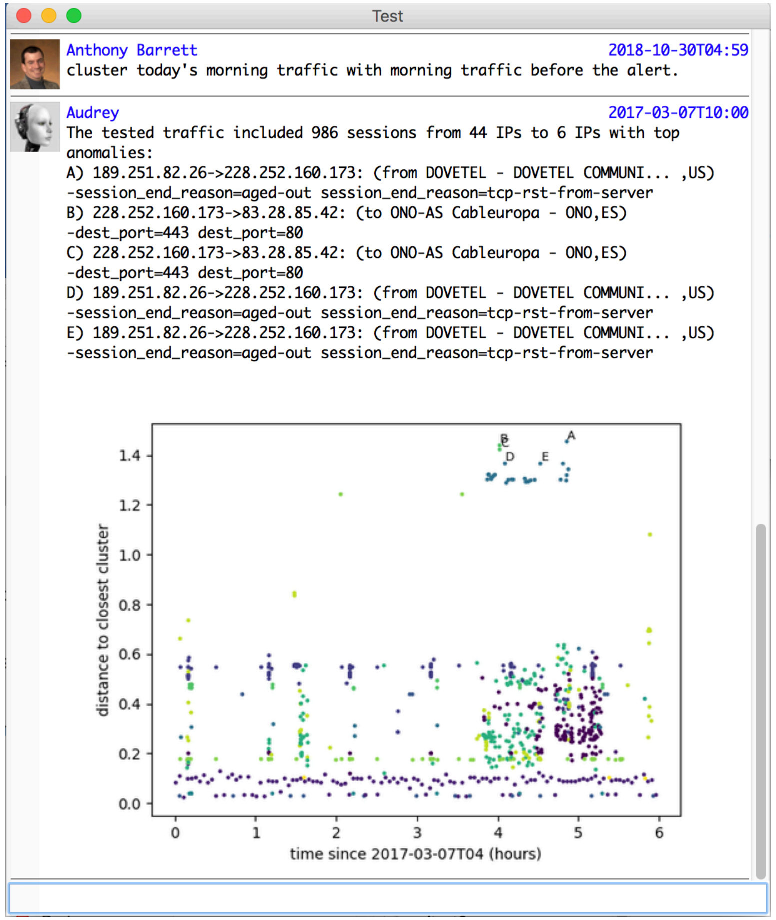
- Underlying intuition of approach
  - Train a compressor on past traffic (assumed safe before alert).
  - A pre-trained compressor's performance degrades on novel new traffic.
  - The objective is to detect novel new traffic.
- Compress/decompress using an autoencoder (AE)

Train on a week of log data prior to the CVE alert to learn baseline behavior that an attack would deviate from.

Apply AE to log data after the CVE alert to find anomalies where error exceeds a threshold, suggesting an attack.



K-means enhancements to analyze logs

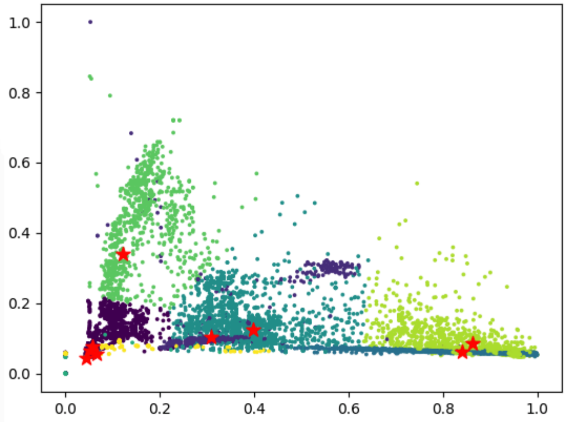


- K-means is an algorithm for partitioning a set of points into a set of K classes, minimizing the distance of a point to its class centroid.
  - Determine classes from past traffic
  - Test new traffic by determining the distance to the closest centroid
- Anomalous traffic is relatively far to its nearest centroid.
  - This is also tracible to symptoms

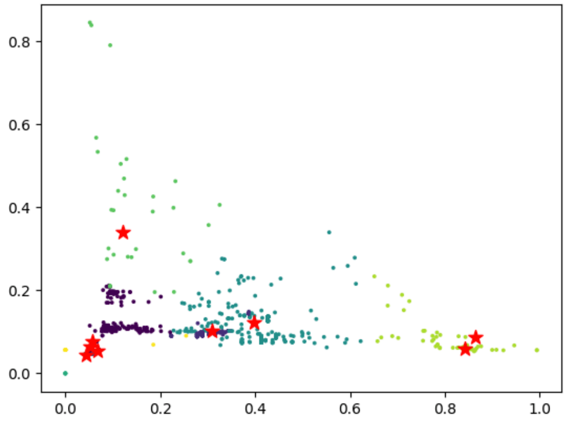
K-Means Algorithm

- Choosing K with elbow method
  - Start with K=1.
  - Increment K until average distance error stops improving by over 10%.
- Example using bytesPerPacketIn vs bytesPerPacketOut scatter plots

Training data before mid-way demo event

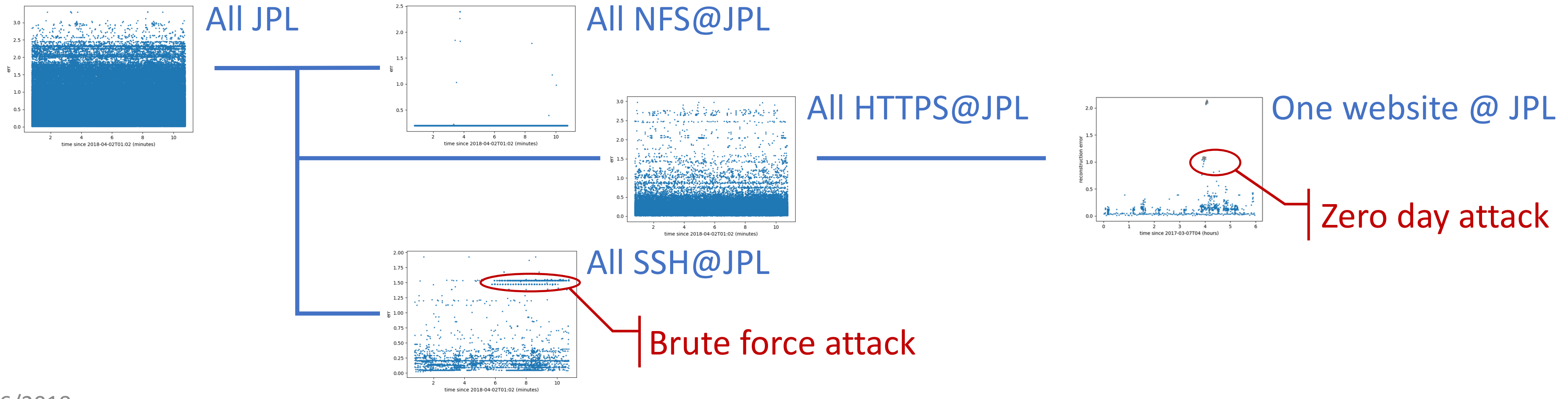


Test data during mid-way demo event



Future work

- Inherently we are just comparing traffic, which has multiple applications: Detecting anomalies, attribution, …
- Adding domain knowledge for automated event exoneration & linked analysis.
- Real-time monitoring
- Some times the comparison is saturated, motivating finding ways to slice traffic into subsets for separate analysis.



Acknowledgements

- Special thanks to David Gilliam for contributing technical discussions.
- This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under NASA prime contract 80NM0018D0004, Task Plan Number 81-19428.